

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods And Arrangements For Providing A Mark-Up
Language Based Graphical User Interface For User
Identification To An Operating System**

Inventors:

Giampiero M. Sierra

&

Christopher A. Evans

1 **RELATED APPLICATION**

2 ~~This application claims priority from U.S. Provisional Application Serial~~
3 No. / , filed December 15, 1999 (Applicant's Docket Number
4 149399.1, entitled "Web-Based User Interface For User Identification To The
5 Operating System", express mailing label number EL425348720US), the
6 ~~disclosure of which is incorporated by reference herein.~~

7
8 **TECHNICAL FIELD**

9 This invention relates to computers and software, and more particularly to
10 methods and arrangements that provide a mark-up language based graphical user
11 interface (GUI) that can be implemented to identify users to an operating system.

12
13 **BACKGROUND OF THE INVENTION**

14 Computer systems are often protected by a logon program and/or other
15 subsequent authentication programs that determine whether a user has permission
16 to access certain computer system resources. By way of example, a networked
17 computer may require that a user input a valid user name and password before the
18 user is allowed to access network resources. Similarly, a Web site on the World
19 Wide Web (WWW) portion of the Internet or on an intranet may require a valid
20 user ID and password before the user is allowed to gain further access to various
21 resources.

22 Controlling user access is not limited to networked computers. A single
23 computer that is accessed by several users may also need to limit access to files
24 and/or various programs therein. Thus, for example, in a home environment, a
25 parent may decide to limit a child's access to the computer entirely, certain

1 programs and/or certain data. Similarly, in a business environment, certain users
2 may have limited access.

3 Controlling access to computers as described above is well known.
4 Typically, there is an initial logon program or the like that requests user input,
5 receives the user input and determines if the user is allowed access. Once the user
6 has been authenticated, then other programs are allowed to operate. For example,
7 in a networked operating system environment, during the booting-up of a personal
8 computer (PC) or like device connected to the network, the user is typically
9 presented with a modal dialog requesting a user name and associated password. In
10 this example, the modal dialog is displayed by the network's logon program. For
11 a single PC, a logon program associated with the operating system may display a
12 similar modal dialog.

13 In either case, the modal dialog tends to be tightly integrated within the
14 logon program code of the network software and/or operating system software.
15 As a result, it is often difficult and expensive to significantly alter the modal
16 dialog or otherwise to introduce new functional and nonfunctional features, such
17 as those typically associated with conventional graphical user interfaces (GUIs).

18 Thus, there is need for improved methods and arrangements that provide
19 enriched techniques for identifying users to an operating system. Preferably, the
20 methods and arrangements will allow for a more advanced GUI to be presented to
21 the user, while also remaining easy for the developer to maintain and modify.

1 **BRIEF DESCRIPTION OF THE DRAWINGS**

2 A more complete understanding of the various methods and arrangements
3 of the present invention may be had by reference to the following detailed
4 description when taken in conjunction with the accompanying drawings wherein:

5 Fig. 1 is a block diagram depicting an exemplary computer system.

6 Fig. 2 depicts an exemplary mark-up language based graphical user
7 interface suitable for use in the computer system of Fig. 1 in identifying users to
8 the operating system.

9 Fig. 3 is a flowchart depicting an exemplary process for identifying users to
10 the operating system using a mark-up language based graphical user interface.

11
12 **DETAILED DESCRIPTION**

13 As shown in Fig. 1, computer 20 includes one or more processors or
14 processing units 21, a system memory 22, and a bus 23 that couples various
15 system components including the system memory 22 to processors 21. Bus 23
16 represents one or more of any of several types of bus structures, including a
17 memory bus or memory controller, a peripheral bus, an accelerated graphics port,
18 and a processor or local bus using any of a variety of bus architectures.

19 The system memory includes read only memory (ROM) 24 and random
20 access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the
21 basic routines that help to transfer information between elements within computer
22 20, such as during start-up, is stored in ROM 24.

23 Computer 20 further includes a hard disk drive 27 for reading from and
24 writing to a hard disk, not shown, a magnetic disk drive 28 for reading from and
25 writing to a removable magnetic disk 29, and an optical disk drive 30 for reading

Computer 20 can operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 50. Remote computer 50 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 20. The logical connections depicted in Fig. 2 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, computer 20 is connected to the local network 51 through a network interface or adapter 156. When used in a WAN networking environment, computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet. Modem 54, which may be internal or external, is connected to bus 23 via interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Reference is now made to Fig. 2, which depicts an exemplary mark-up language based graphical user interface (GUI) display suitable for use in identifying users to the operating system of computer 20. Here, a logon screen 100 is shown as having a first title area 102 that identifies logon screen 100. Logon screen 100 may take up be a full screen of display 47 or a portion thereof. As shown, first title area 102 can include any graphical feature (e.g., bitmap, photo, video, text, etc).

Process 200 can be implemented, for example, within a Microsoft Windows operating system environment using Dynamic HTML and available interfaces. Thus, a logon process, known as WinLogon, spawns a separate process to host the Dynamic HTML content. When WinLogon launches the separate process, it provides a mechanism to communicate with WinLogon so that the HTML interface can ask WinLogon to authenticate the user and start their desktop session. The Dynamic HTML code then makes calls to an ActiveX control or like applet that communicates with the operating system to determine the list of users, the picture to associate with the user, a password hint if one was configured by the user and whether the user has a password configured. The user then selects their picture or name, for example, and types in their password (if needed), after which the HTML code calls the ActiveX control with the user name and password. The ActiveX control then passes this information back to WinLogon where the authentication takes place. If the user is authenticated, then WinLogon creates the user's desktop and switches to it. At this point, the Dynamic HTML process is finished. Thereafter, the rendering engine may remain loaded or may be terminated.

Although some preferred embodiments of the various methods and arrangements of the present invention have been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the exemplary embodiments disclosed, but is capable of numerous rearrangements, modifications and substitutions without departing from the spirit of the invention as set forth and defined by the following claims.